Hindawi Publishing Corporation
EURASIP Journal on Image and Video Processing
Volume 2010, Article ID 183605, 15 pages
doi:10.1155/2010/183605

Research Article

Robust, Real-Time 3D Face Tracking from a Monocular View

Wei-Kai Liao, Douglas Fidaleo, and Gerard Medioni

Department of Computer Science, University of Southern California, Los Angeles, CA 90089, USA

Correspondence should be addressed to Gerard Medioni, medioni@usc.edu

Received 15 March 2010; Revised 7 September 2010; Accepted 4 October 2010

Academic Editor: Peter Eisert

This paper addresses the problem of 3D face tracking from a monocular view. Dominant tracking algorithms in current literature can be classified as intensity-based or feature-based methods. Intensity-based methods track 3D faces based on the brightness constraint, assuming constant intensity of the face across adjacent frames. Feature-based trackers use local 2D features to determine sparse pairs of corresponding points between two frames and estimate 3D pose from these correspondences. We argue that using either approach alone neglects valuable visual information used in the other method. We therefore propose a novel hybrid tracking approach that integrates multiple visual cues. The hybrid tracker uses a nonlinear optimization framework to incorporate both feature correspondence and brightness constraints, and achieves reliable 3D face tracking in real-time. We conduct a series of experiments to analyze our approach and compare its performance with other state-of-the-art trackers. The experiments consist of synthetic sequences with simulated environmental factors and real-world sequences with estimated ground truth. Results show that the hybrid tracker is superior in both accuracy and robustness, particularly when dealing with challenging conditions such as occlusion and extreme lighting. We close with a description of a real-world human-computer interaction application based on our hybrid tracker.

1. Introduction

3D face tracking is a fundamental component for many computer vision problems and forms the basis of many face-related applications. For example, in human-computer interaction, 3D pose is used to determine the user's attention and the mental status. For face and expression recognition, the 3D head pose is required for stabilizing the face as a preprocess. The estimated pose can also assist in 3D face reconstruction from a monocular camera.

In real-world applications, tracking accuracy, computational efficiency, and the robustness of the tracker are all important factors. For real-time or interactive applications, the tracker must be computationally efficient. Robustness can be defined in several ways including robustness to noise, stability on textureless video, insensitivity to illumination changes, and resistance to the expression changes or other local-nonrigid deformation. The tracker should also run continuously for long sequences, requiring a mechanism to prevent drift and error accumulation.

In this paper, we propose a hybrid tracker for 3D face tracking. Instead of relying on any single channel of information, the hybrid tracker integrates distinct, but complementary, visual cues. This idea is inspired by a detailed comparisons between two existing state-of-the-art head trackers [1, 2]. Feature-based methods such as [1, 3] depend on the ability to detect and match the same features in subsequent frames and keyframes. The quantity, accuracy, and face coverage of the matches fully determine the recovered pose quality. In contrast, intensity-based methods such as [2] do not explicitly require feature matching, but expect brightness consistency between the same image patches in different frames to compute the implicit flow of pixels. These two methods are extensively examined in our experiments. Empirical observation suggests that neither tracker is consistently better than the other; each tracker has its own strengths but also its own weaknesses. Thus, by design, the hybrid tracker is expected to overcome the flaws of the single-channel trackers while retaining their strengths. This is clearly demonstrated in our experiments.

The rest of this paper is organized as follows: we start with a literature review of related work in Section 2. Next, Section 3 discusses each of the intensity- and feature-based 3D head-tracking approaches and compares their difference. Based on empirical observation, a hybrid tracking algorithm is proposed. The details of this algorithm are illustrated in Section 4. The proposed hybrid tracker, along with the intensity- and feature-based trackers, are examined thoroughly in various experiments. The results are presented in Section 5. Finally, a summary and conclusions are given in Section 6.

2. Previous Work

The performance of face tracking is affected by many factors. While higher level choices such as whether or not to use keyframes, how many to use, and whether to update them online can alter the accuracy and speed of the tracker; a more fundamental issue is the optimization algorithm and the related objective functional. Most state-of-the-art 3D face-tracking algorithms are affected by the following three factors.

(i) Prior knowledge of the approximate 3D structure of the subject's face. In [4], Fidaleo et al. have shown that the accuracy of the underlying 3D model can dramatically affect the tracking accuracy of a feature-driven tracker. Much of the performance difference between tracking methods can be attributed to the choice of model: planar [5], ellipse [6–8], cylinder [2, 9], and generic face or precise geometry [1]. The 2D planar approximation is very simple, but its lack of 3D structure introduces error in cases of out-of-plane rotation. A 3D ellipsoid or cylinder is often used as an approximation of a human head. Alignment of such geometry is relatively easy due to the simplicity of the models. Precise facial geometry with good initial alignment attains the best performance, but acquisition and subsequent alignment of this data is challenging. When the alignment degrades, the tracking accuracy drops dramatically.

(ii) Observed data in the 2D image. The tracker relies on this information to estimate the head pose. This includes feature locations [1, 3], intensity values in a region [2, 7–10], or estimated motion flow fields [6, 11].

(iii) The computational framework. These can be roughly divided into deterministic and stochastic methods [12]. For deterministic methods, an error function is defined using the observed 2D data and the corresponding estimated 2D data. Pose parameters are adjusted to minimize this error function. Most of the deterministic methods use a nonlinear optimization approach, which relies on the gradient-based method such as Gaussian-Newton or Levenberg-Marquardt. The scheme adopted (line search or trust region) and the method to compute first- and second-order derivatives highly affect the convergency, efficiency, and accuracy of the method. On the other hand, stochastic estimation methods such as particle filtering (sequential Monte Carlo) and Markov Chain Monte Carlo define the observation and transition models for tracking. Model fitness and the quality of the estimated model parameters determines the tracking accuracy whereas the efficiency depends on the model complexity and choice of filtering algorithm. In general, deterministic methods are more computational efficient, while stochastic methods are more resistant to the local minima.

3. Intensity-versus Feature-Based Tracking

This section compares the intensity- and feature-based tracking methods. To prepare the readers, we first review the individual algorithms. The selected representative algorithm for each class is [1, 2] for the intensity- and feature-based methods, respectively. The fundamental concepts of these trackers are summarized, and the reader is referred to the original papers for the specific details.

3.1. Intensity-Based Trackers. The intensity-based tracker performs optimization based on the brightness constraint. To be more specific, let $\mu = \{t_x, t_y, t_z, \omega_x, \omega_y, \omega_z\}^T$ be the motion vector specifying the 3D head pose. Given the pose in frame $t-1$, μ_{t-1}, we define an error function $E_t(\Delta\mu)$ for $\Delta\mu$, the incremental pose change between frame $t-1$ and t, as

$$E_t(\Delta\mu;\mu_{t-1}) = \sum_{p\in\Omega} \left\| I^{t-1}(F(p,0;\mu_{t-1})) - I^t(F(p,\Delta\mu;\mu_{t-1})) \right\|_2^2, \quad (1)$$

here, Ω is the face region, and p is the 3D position of a point on the face. $F = P \circ M$, where $M(p,\Delta\mu)$ will transform the 3D position of p as $\Delta\mu$ specified and P is a weak perspective projection. $I^t(\cdot)$ and $I^{t-1}(\cdot)$ are the frame t and $t-1$, respectively.

This error function measures the *intensity difference* between the previous frame and the transformed current frame. If the intensity consistency is maintained and the noise of intensity is Gaussian distributed, the minimum of this 2-norm error function is guaranteed to be the optimal solution. Thus, by minimizing this error function with respect to the 3D pose, we can estimate the change of 3D pose and recover the current pose.

Offline information can also be integrated into the optimization similar to Vacchetti et al. [1]. The error function $E_k(\Delta\mu)$

$$E_k(\Delta\mu;\mu_{t-1}) = \sum_{i=1}^{N_k} \alpha_i \left[\sum_{p\in\Omega} \left\| I^i(F(p,0;\mu_i)) - I^t(F(p,\Delta\mu;\mu_{t-1})) \right\|_2^2 \right] \quad (2)$$

is defined between the current frame and the keyframes. N_k is the number of keyframes. $I^i(\cdot)$, and μ_i are the frame and pose of the ith keyframe. This error function can use both offline or online generated keyframes for estimating the head pose.

A regularization term

$$E_r(\Delta\mu;\mu_{t-1}) = \sum_{p\in\Omega} \left\| F(p,0;\mu_{t-1}) - F(p,\Delta\mu;\mu_{t-1}) \right\|_2^2 \quad (3)$$

can also be included to impose a smoothness constraint over the estimated motion vector.

The final error function for optimization is the combination of (1), (2), and (3)

$$E_{\text{int}} = E_t + \lambda_k E_k + \lambda_r E_r, \tag{4}$$

where λ_k and λ_r are weighting constants. This is a nonlinear optimization problem, and the iteratively reweighted least square is applied.

3.2. Feature-Based Trackers. The feature-based tracker minimizes the reprojection error of a set of 2D and 3D points matched between frames. A keyframe in [1] consists of a set of 2D feature locations detected on the face with a Harris corner detector and their 3D positions estimated by back-projecting onto a registered 3D tracking model. The keyframe accuracy is dependent on both the model alignment in the keyframe image, as well as the geometric structure of the tracking mesh. These points are matched to patches in the previous frame and combined with keyframe points for pose estimation.

The reprojection error for the keyframe feature points is defined as

$$E_{k,t} = \sum_{p\in k} \left\| m_t^p - F(p,\mu_t) \right\|_2^2, \tag{5}$$

where κ is the set of keyframe feature points, m_t^p is the measured 2D feature point corresponding to the keyframe feature point p at frame t, and $F(p,\mu_t)$ is the projection of p's 3D position using pose parameters μ_t.

To reduce jitter associated with single-keyframe optimization, additional correspondences between the current and previous frame are added to the error term

$$E_t = \sum_{p\in k} \left(\left\| n_t^p - F(p,\mu_t) \right\|_2^2 + \left\| n_{t-1}^p - F(p,\mu_{t-1}) \right\|_2^2 \right), \tag{6}$$

where the 3D locations for the new points is estimated by back projection to the 3D model at the current pose estimate.

The two terms are combined into the final error functional

$$E_{\text{fpt}} = E_{k,t} + E_{k,t-1} + E_t, \tag{7}$$

which is minimized using nonlinear optimization.

3.3. Comparison. Both tracking methods are model based, using an estimate of the 3D shape of the face and its projection onto the 2D image plane to define a reprojection error functional that is minimized using a nonlinear optimization scheme. The forms of the error functionals are nearly identical, differing only in the input feature space. Figure 1 illustrates the difference between these 2 trackers.

For the feature-based tracker, the reprojection error is measured as the *feature distance* between a set of key 2D features and their matched points in the new image. The tracker relies on robust correspondence between 2D features in successive frames and keyframes, and thus the effectiveness of the feature detector and the matching algorithm is critical for the success of the tracker. In [1], Vacchetti et al. use the standard eigenvalue-based Harris corner detector. Using a more efficient and robust detector should improve the feature-based tracker.

In contrast, the intensity-based tracker utilizes the brightness constraint between similar patches in successive images and defines the error functional in terms of *intensity differences* at sample points.

To determine the role of this input space on tracking accuracy, we perform a set of controlled experiments on synthesized motion sequences (see Section 5 for details). Feature-based methods are generally chosen for their stability under changing or extreme lighting and other conditions, with the assumption that feature locations remain constant despite these changes. For cases where there is insufficient texture on the face (low resolution, poor focus, etc.), the accuracy of feature methods quickly degrades. Intensity-based methods are more widely applicable and can perform well in low- or high-texture cases, however, they are clearly sensitive to lighting changes. We demonstrate this empirically by testing on the near-infrared sequence. Both tracking methods have difficulty in the case of occlusion and often resort to offline information (keyframes) and/or statistical outlier estimation for robustness. We show that by reformulating the tracking problem to harness both feature types, we can improve robustness in all tested scenarios.

4. Our Hybrid Tracker

The empirical and theoretical comparison of intensity- and feature-based tracker inspires the design of our hybrid-tracking algorithm. In this section, we reformulate the 3D face-tracking problem as a multiobjective optimization problem and present an efficient method to solve it. The robustness of the tracker is also discussed.

4.1. Integrating Multiple Visual Cues. Integrating multiple visual cues for face tracking can be interpreted as adjusting the 3D pose to fit multiple constraints. The hybrid tracker has two objective functions with different constraints to satisfy simultaneously (4) and (7). This becomes a multiobjective optimization problem. Scalarization is a common technique for solving multiobjective optimization problems. The final error function becomes a weighted combination of the individual error functions (4) and (7)

$$E = a_i E_{\text{int}} + a_f E_{\text{fpt}}, \tag{8}$$

where a_i and a_f are the weighting constants.

The hybrid tracker searches for the solution to minimize (8). The process can be interpreted as a nonlinear optimization based on brightness constraints, but regularized with feature correspondence constraints. Ideally, these two constraints compensate for each other's deficiencies. The feature point correspondences restrict the space of feasible solutions for the intensity-based optimization and help the optimizer to escape from local minima. The brightness

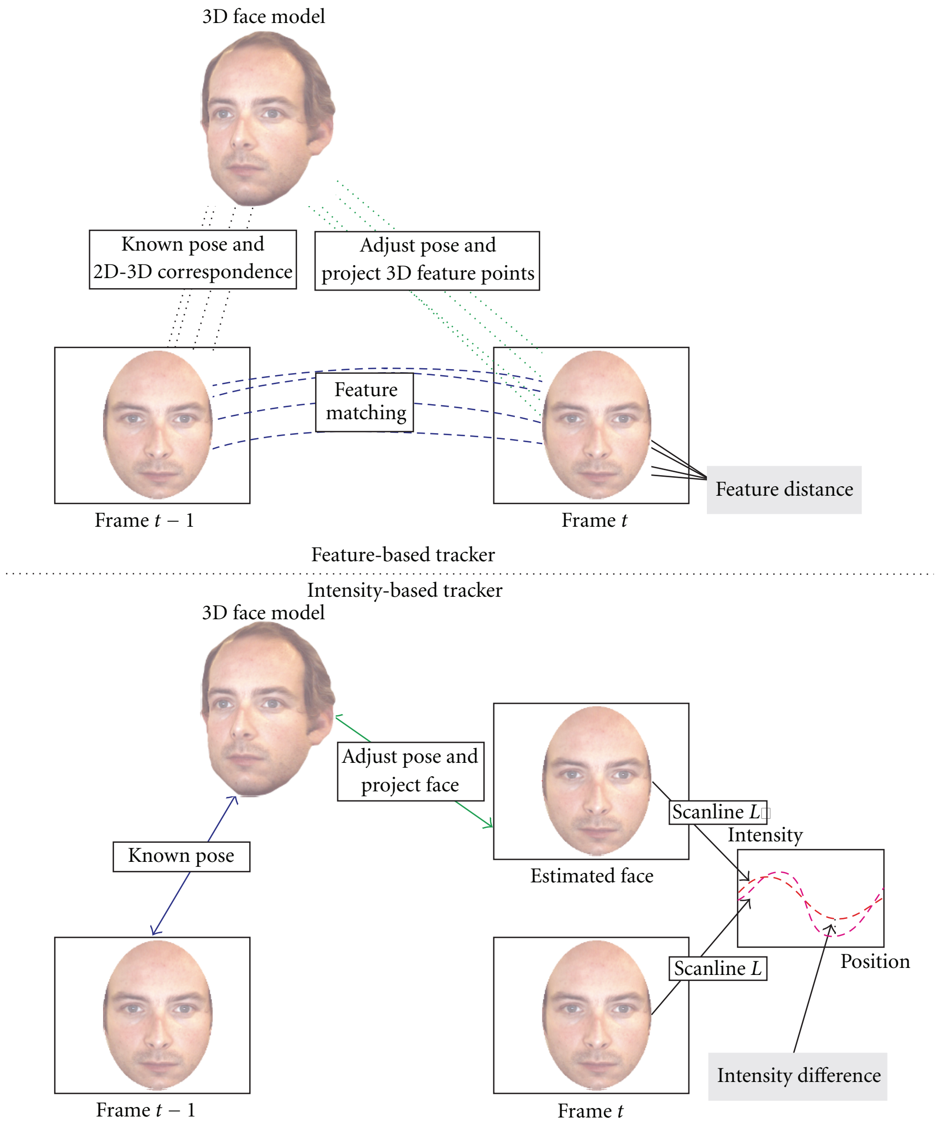


FIGURE 1: Difference in optimization source data for the feature-based tracker, T_F, and the intensity-based tracker, T_I. Given a set of key feature points defined on a 3D model, and their projection, T_F minimizes the total distance to matched feature points in pixel space. T_I computes the pose that minimizes the total intensity difference of pixels under the feature points.

constraint, on the other hand, refines and stabilizes the feature-based optimization. When there are not sufficient high-quality feature matches, the intensity constraint still provides adequate reliable measurement for optimization.

The convergence of feature-based optimization is much faster than intensity-based methods due to the high dimensionality of the image data and the nature of the associated imaging function. However, when E_{fpt} is close to its optimum, E_{int} still provides information to refine the registration. Therefore, an adaptive scheme is applied to choose the weights a_i and a_f. At the beginning of the optimization, E_{fpt} has higher weight and decreases when it approaches its optimum. Meanwhile, the weight of E_{int} becomes more important when the optimization proceeds. The overall distribution of the weights is also affected by the number of matched features. In the case of few feature correspondences, the tracker reduces the weight of E_{fpt}

$$a_f = \begin{cases} \dfrac{a_f^0}{(\text{iter}+1)} & \text{featurenumber} \geq n_2, \\ \dfrac{a_f^0}{(\text{iter}+1)} \times c, \quad 0 < c < 1, & \\ \quad n_1 \leq \text{featurenumber} < n_2, & \\ 0 & \text{featurenumber} < n_1, \end{cases}$$

$$a = a_f + a_i, \tag{9}$$

where a is the constant value for summation of a_f and a_i, c is a constant ratio to reduce a_f, a_f^0 is the constant initial feature weight at the first iteration of the optimization, and *iter* is the current iteration number. n_1 and n_2 are the thresholds to take matched feature number into account, and $0 < n_1 \le n_2$.

4.2. Efficient Solution. The computational cost of the feature-based tracker is low due to the relatively small number of matched features and the fast convergence of the optimization. On the other hand, intensity-based trackers are notorious for their high computational cost. The standard algorithm for solving this iterative least-square problem is slow, due to the evaluation of a large Jacobian matrix $F_\mu = \partial F / \partial \mu$ and approximation to Hessian matrix $(I_u F_\mu)^T (I_u F_\mu)$, where I_u is the gradient of the frame I. This can be accelerated using the (forward) compositional algorithm, but the evaluation of the Hessian is still required at each iteration.

To improve the overall tracking speed, we use the inverse compositional algorithm as proposed by Baker and Matthews for image alignment in [13]. In the inverse compositional algorithm, the Jacobian and Hessian matrices are evaluated in a preprocessing step, and only the error term is computed during the optimization. The face image is warped at each iteration, and the computed transform is inverted to compose with the previous transform. Here, warping the image is equivalent to model projection. Since we know the 2D-3D correspondence in I_{t-1}, warping I_t for intensity difference evaluation is achieved by projecting the 3D model and sampling to get the intensity in I_t.

The inverse compositional version of our algorithm is following.

(i) Preprocess:

  (a) for E_{int}: compute the gradient image, the Jacobian, and the Hessian matrix,

  (b) for E_{fpt}: perform feature detection on I_t, and feature matching between I_t, I_{t-1}, and keyframes.

(ii) Optimization
At each iteration:

  (1) For E_{int}:

    (1.1) warp the face region of I_t to get the intensity,

    (1.2) compute the intensity difference and the weight.

  (2) For E_{fpt}:

    (2.1) project the feature points to get the 2D position,

    (2.2) compute the reprojection error and weights.

  (3) Solve the linear system.

  (4) Update the pose.

(iii) Postprocess: back-project the face region and feature points of I_t into the 3D face model.

4.2.1. Practical Considerations. Though the inverse compositional (IC) algorithm is frequently used as the default algorithm for solving for 3D object pose in intensity-based 3D tracking, it is not mathematically equivalent to the forward compositional (FC) algorithm as discussed in [14]. In general, the preprocessing time for IC is longer than that for FC; therefore, in cases where the iteration number is small (fast convergence due to low interframe variation) or the analyzed face-region is small, the benefit of IC over FC is less apparent. However, as the face region size or the iteration number increases, the benefit of the inverse compositional algorithm becomes clear, since each iteration takes significantly less time. In largely unconstrained scenarios such as ours, the IC approach provides a good balance between accuracy and performance.

4.3. Local Features. We adapt the SIFT (Scale Invariant Feature Transform) [15] detector to extract 2D local features for the hybrid tracker. Feature matching is performed by searching for the candidate with minimum 2-norm distance of feature descriptor [15]. To reject false matches, we require a large feature distance between two top candidate matches

$$\|x - x_1\| < \alpha \times \|x - x_2\|, \tag{10}$$

where x is the input feature point descriptor, and x_1 and x_2 are the best and the second-best keypoint candidates, respectively. α is the threshold defines the distance ratio for rejection.

To further reduce outliers of correspondence pairs, we also exploit the heuristic that in the presence of small interframe motion, two corresponding points must be spatially close to each other. Given two sets of SIFT features, $\{x_m\}$ and $\{y_n\}$, two keypoints $x_i \in \{x_m\}$ and $y_j \in \{y_n\}$, (x_i, y_j) is considered as a correspondence pair if

$$\begin{gathered} y_j = \text{FeatureMatch}(x_i, \{y_m\}), \\ x_i = \text{FeatureMatch}\left(y_j, \{x_n\}\right), \\ \left\| P_{x_i} - P_{y_j} \right\|_2 < d, \end{gathered} \tag{11}$$

where "FeatureMatch" is the matching approach presented in previous paragraph, and P_{x_i} and P_{y_j} are the 2D coordinate of x_i and y_j in the image plane. d is a real value threshold to impose the "closeness" constraint.

4.4. Practical Considerations. SIFT analyzes features at multiple octaves using a DoG (difference of Gaussian) approximation to the LoG (Laplacian of Gaussian). These computations are nontrivial and must be repeated for each octave to achieve true scale invariance. However, for the 3D head-tracking scenario described in this paper, the size variation between two face adjacent images is not significant. Therefore, we reduce overhead in the SIFT computation by restricting analysis to a single octave.

Figure 2 compares feature detection and matching results of the full SIFT analysis and our simplified single-octave SIFT on a face in two consecutive frames. In this example, there

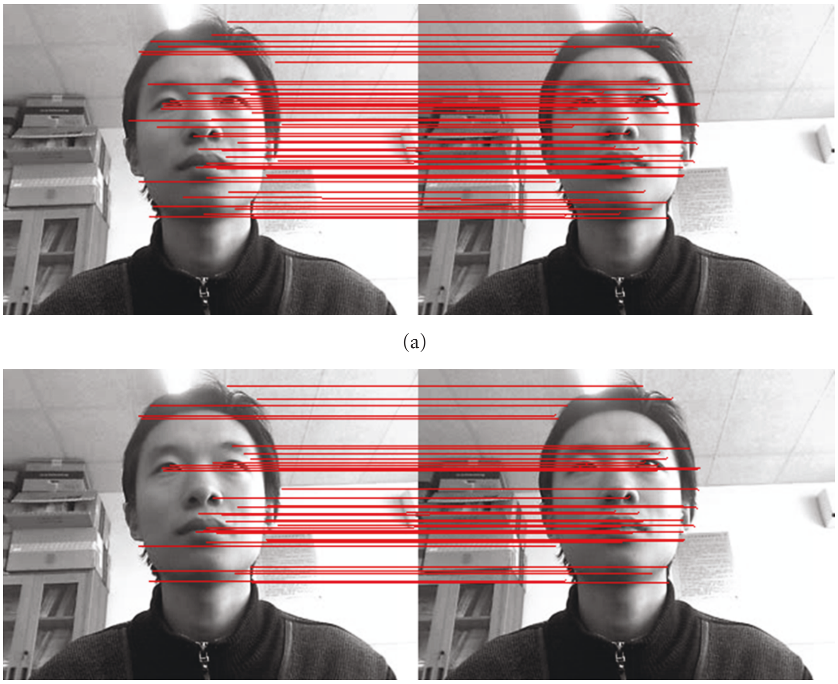


Figure 2: Comparison of full SIFT and simplified SIFT for face tracking. The top row shows the corresponding feature points from full SIFT detector, and the bottom row is from the simplified SIFT detector. See Section 4.3 for more details.

are 70 matched feature points when using full SIFT, while there are only 51 matched features for the simplified SIFT detector. The decrease in match count is acceptable for our experiments given real-time constraints.

5. Experiments

A series of face tracking evaluations are performed. The first set of experiments uses synthetic sequences. Using synthetic sequence guarantees, the exact ground truth is available. We have full control over sequence generation, and thus can isolate each factor and test the tracker's response. The next experiment tests the performance of the tracker in real video sequences. The collected video sequences and one public benchmark database are used for evaluation. In a third experiment, we test the performance on textureless videos. We have a real-world application that demands the use of a near-infrared camera. The face tracker is used to extract head pose for human-computer interaction. We present tracking results of the proposed hybrid tracker in this challenging setting. In these experiments, the proposed tracker and the existing state-of-the-art tracking algorithms are evaluated and compared. The feature-based tracker is an implementation of [1]. The intensity-based and hybrid tracker are C++ implementations of the methods presented in Sections 3 and 4.

5.1. Evaluation with Synthetic Sequences

5.1.1. Experimental Setup. The proposed hybrid tracker, the intensity-, and feature-based tracker are evaluated on synthetic sequences of four subjects. All trackers use a precise 3D face model acquired with the FaceVision modeling system [16] to rule out the effects of model misalignment. For each model, three independent sequences of images are rendered. The first consists of pure rotation about the x-(horizontal) axis, the second is rotation about the y-(vertical) axis, and the third is rotation about the z-axis. In each case, the sequences begin with the subject facing the camera and proceed to -15 degrees, then to 15 degrees, and return to neutral in increments of 1 degree. A total of 60 frames are acquired for each sequence. The image size is 640×480.

Synthetic perturbations are applied to the sequences to mimic variations occurring due to lighting, occlusion, and facial deformation changes. The following test configurations are used to evaluate the tracking performance.

Ambient. The models are rendered with constant ambient lighting. This removes all factors influencing the tracking accuracy.

Diffuse. The models are rendered with a strong single directional light source using a Lambertian reflectance model. This is a challenging test case with extreme lighting.

Specular. The models are rendered with a strong single directional light source using a Phong reflectance model. This adds mobile highlights and also presents a challenging test case.

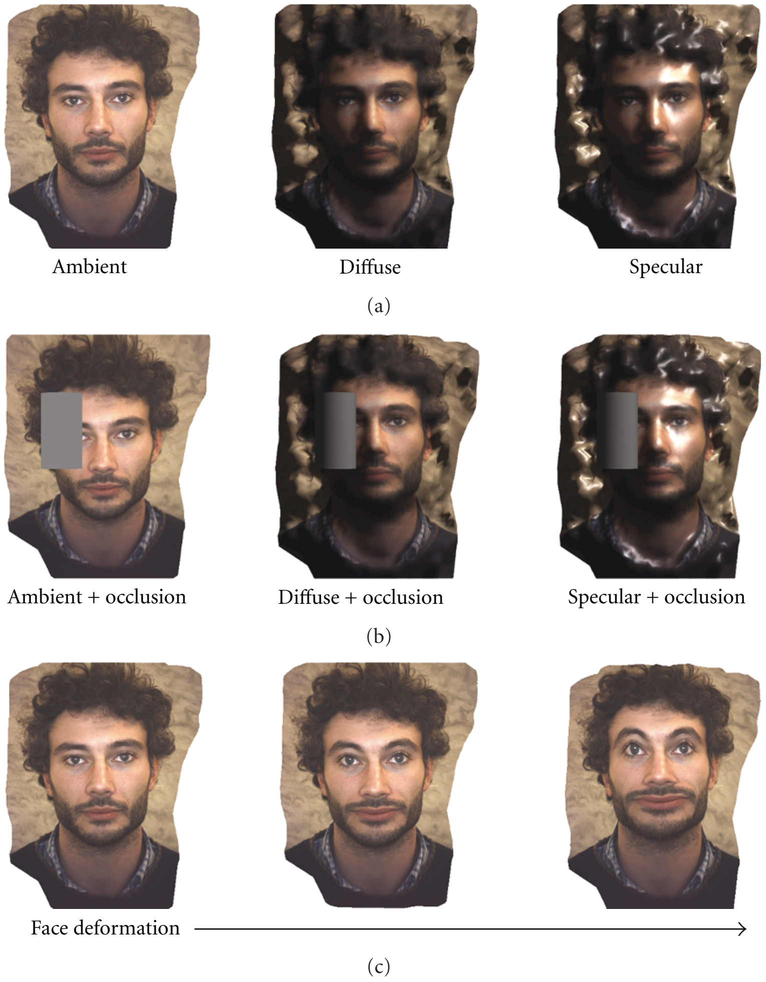


Figure 3: Example synthetic sequences used for experiments. (a) sequences with variable lighting/material conditions. From left to right: ambient, diffuse, and diffuse+specular. (b) same lighting conditions above with added occlusion from an animated cylinder passing between the camera and subject. (c) deformation with a simple face muscle system.

Occlusion. The three lighting cases above are repeated with the addition of a small opaque cylinder moving slowly across the face.

Deformation. A synthetic muscle system is used to deform the face mesh over the course of the sequence. The muscles are contracted at a constant rate over the duration of the sequence inducing deformation in the mouth and eyebrow region (two high-texture areas on the face). The model is rendered with constant ambient lighting.

Figure 3 shows some examples from the synthetic sequences. The faces in the rendered sequences have a large amount of surface texture and are, therefore, amenable to feature based tracking.

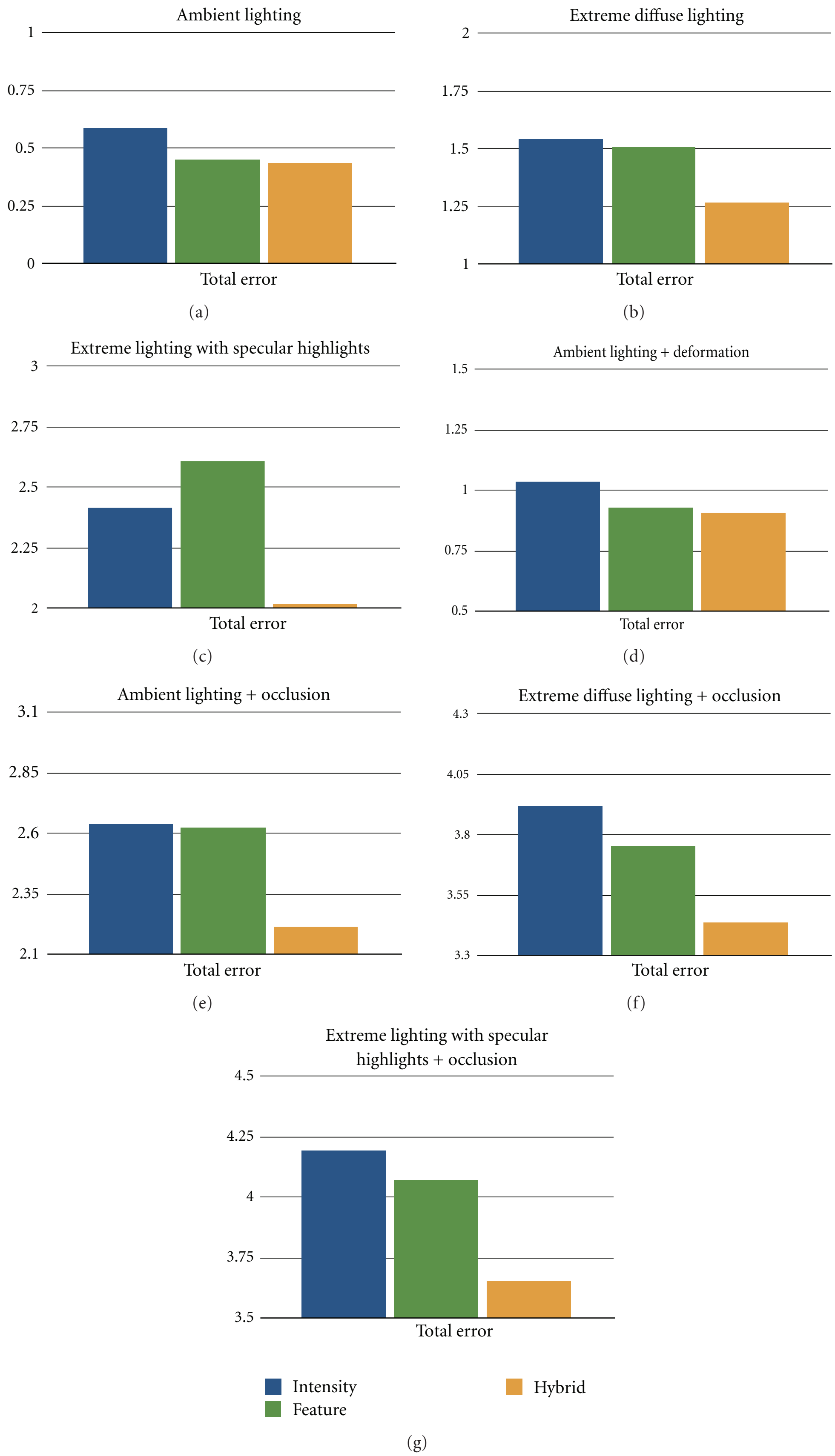
5.1.2. Analysis. Figure 4 shows the error of each tracker computed as a sum of the mean-squared rotational error over all dimensions and frames. This error measures the absolute difference between the estimated angle and the true angle. In this evaluation, the average speed of the proposed tracker is 30 frame-per-second (FPS) on a normal desktop with one Intel XEON 2.4 GHz processor.

The hybrid tracker consistently outperforms the other trackers. In the cases of lighting variation, the hybrid tracker is only marginally better than the second best method. However, the hybrid tracker show considerable improvement in the presence of occlusion.

Ambient. All trackers perform well, despite the different optimization functionals. The hybrid tracker exhibits marginal improvements.

Diffuse. As the head moves, the appearance of the face changes dramatically due to the extreme lighting conditions. This presents considerable challenges for the intensity-based tracker. As expected, the intensity-based tracker's performance degrades significantly. The hybrid method is comparable (slightly better) than the feature tracker.

Specular. This scenario is identical to the diffuse case with the addition of specular highlights that move across the face



(a) (b) (c) (d) (e) (f) (g)

Figure 4: Average error for synthetic sequences. Each figure plots the averaged error over all four subjects. Errors for x-, y-, and z-axis angle are aggregated.

as the head rotates. This presents an additional challenge for the feature tracker as any features detected at the highlight boundary may incorrectly bias the tracker estimate. Furthermore, highlights are defined by regions of high color saturation which provide little information to either tracking method. The hybrid tracker performs considerably better than the other two trackers appearing to compensate for the errors introduced by either tracker independently.

Deformation. All trackers perform worse than the optimal cases, but the accuracy is still acceptable. As deformation increases with time the accuracy of all methods declines. The intensity-based method is only slightly worse than the feature-based method, since the usage of the region-based difference compensates for the outliers and improves the robustness.

Occlusion. The strengths of the hybrid tracker are evident in the three occlusion cases. While both the intensity and feature trackers exhibit similar performance, the hybrid tracker consistently achieves better accuracy. This indicates that some portion of the error introduced by the two unimodal trackers is orthogonal. By merging the information, we improve the robustness of the independent trackers.

5.2. Evaluation with Real Sequences. The proposed tracker is also evaluated with many real sequences. One problem of evaluating with real sequences is the lack of ground truth. Only "estimated ground truth" is available. In the literature, several methods are used to estimate the ground truth, such as with a magnetic tracker or offline bundle adjustment. We perform the evaluation with two different sets of sequences. One is collected in our lab, and the other is from the Boston University (BU) database [9].

The BU database contains 2 sets of sequences: uniform lighting and varying lighting. The uniform lighting class includes 5 subjects, totalling 45 sequences. Figure 5 shows the tracking result of the "jam5.avi" sequence in the uniform lighting class. Overall, the estimated pose is close to ground truth. Note that the apparent jitter in the ground truth graph is due to noise from the magnetic tracker.

Our sequence is captured in an indoor environment. The ground truth is estimated by commercial bundle adjustment software [17]. These sequences contain large rotations with a maximum angle near 40 degrees. The hybrid tracker tracks the 3D pose reliably. Figure 6 shows the tracking result of one sequence.

Figure 7 shows the comparison of the hybrid tracker and the intensity-based tracker in a strong reflection case. The intensity-based tracker is sensitive to lighting change, since it violates the brightness consistency assumption. In Figure 7, there is a strong reflection on the subject's forehead, and it moves as the subject turns his head. As shown in the figure, the drift of intensity-based tracker is much larger than the hybrid tracker, especially for the pose is far away from the frontal view (see the third and forth column of Figure 7). This example clearly demonstrates the robustness of the hybrid tracker.

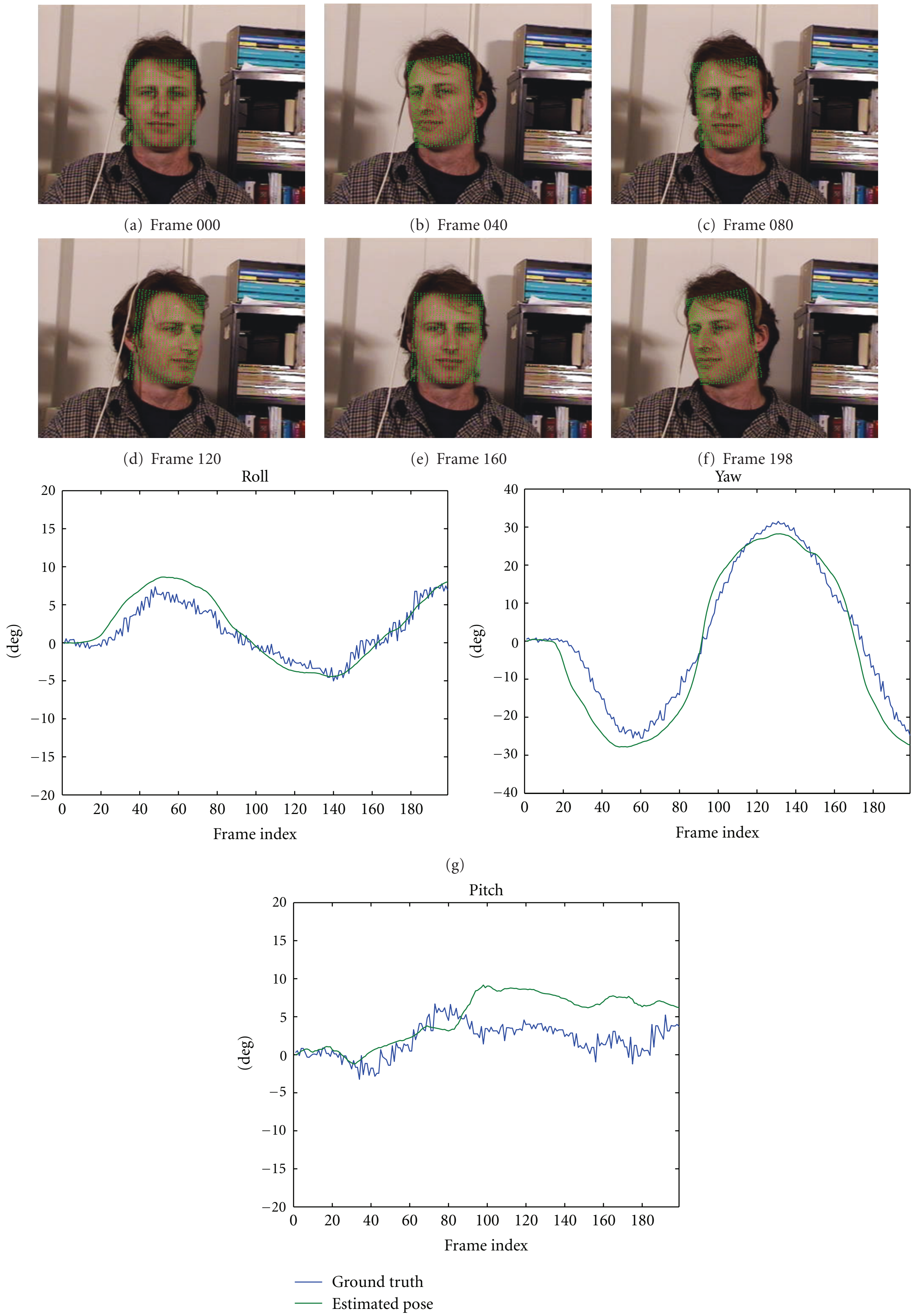
5.3. Infrared Sequences and Application. Infrared (IR) images are commonly used in vision applications in environments where visible light is either nonexistent, highly variable, or difficult to control. Our test sequences are recorded in a dark, theater-like interactive virtual simulation training environment. In this environment, the only visible light comes from the reflection of a projector image off a cylindrical screen. This illumination is generally insufficient for a visible light camera and/or is highly variable. The tracker estimates the head pose, indicating user's attention and is used in a multimodal HCI application. The theater environment and sample IR video frames are shown in Figure 8. Ground truth is not available for this data; therefore, only qualitative evaluation is made.

IR light is scattered more readily under the surface of the skin than visible light. Microtexture on the face is therefore lost (especially at lower resolution), making identification of stable features more difficult and error prone. Due to varying absorption properties in different locations of the face, however, low-frequency color variations will persist which satisfy the brightness constraint.

Figure 9 shows the tracking results in this environment. It shows multiple frames across a several minute sequence. The video is recorded at 15 FPS, and its frame size is 1024 × 768. In most cases, the face size is around 110 × 110. The subject's head moves in both translation and rotation. There are also some mild expression changes (mouth open and close), and strong reflection in some frames. In this experiment, the user is assumed to begin in a frontal view. The tracker uses only one keyframe, the first frame. No offline training is involved. The proposed hybrid tracker reliably tracks the pose in real-time with large head motion, while the feature-based tracker loses track completely after only 3 frames. Probing deeper, we see that when feature-based tracker is lost, only a few features (1–4) are reliably matched on each frame. This exemplifies the problem with feature-based methods on low-texture images.

Another interesting observation is related to error accumulation. In Figure 9, the center column shows a frame with strong reflection coming from the subject's glasses. At that frame, the tracking accuracy degrades, due to the insufficient number of the features matched in this environment. However, after the reflection disappears, the tracker recovers. This demonstrates how the use of keyframes prevents error accumulation.

5.4. Practical Considerations: Automatic Initialization and Reacquisition. Automatic initialization and reacquisition are critical for using 3D head tracker in real-world applications. The initialization affects the performance of a tracker, and the automatic requisition module enables the tracker recovering from lost track. In this work, we make an assumption that the tracker only initializes and reacquires the face in the frontal pose. We use the face detector from [18] to locate a frontal face. In the tracker initialization stage, the face detector searches the entire frame for a frontal face. If the detector consistently locates a face near a certain position, the 3D head model is fitted into the detected face area by changing its 3D position. To improve the accuracy



(a) Frame 000

(b) Frame 040

(c) Frame 080

(d) Frame 120

(e) Frame 160

(f) Frame 198

(g)

(h)

Figure 5: Evaluation on the BU database. The top rows show some examples from the tracker, and the last row shows the estimated roll, yaw, and pitch compared with the ground truth from magnetic tracker. The result is for the "jam5.avi" sequence in the uniform lighting class of the BU database.

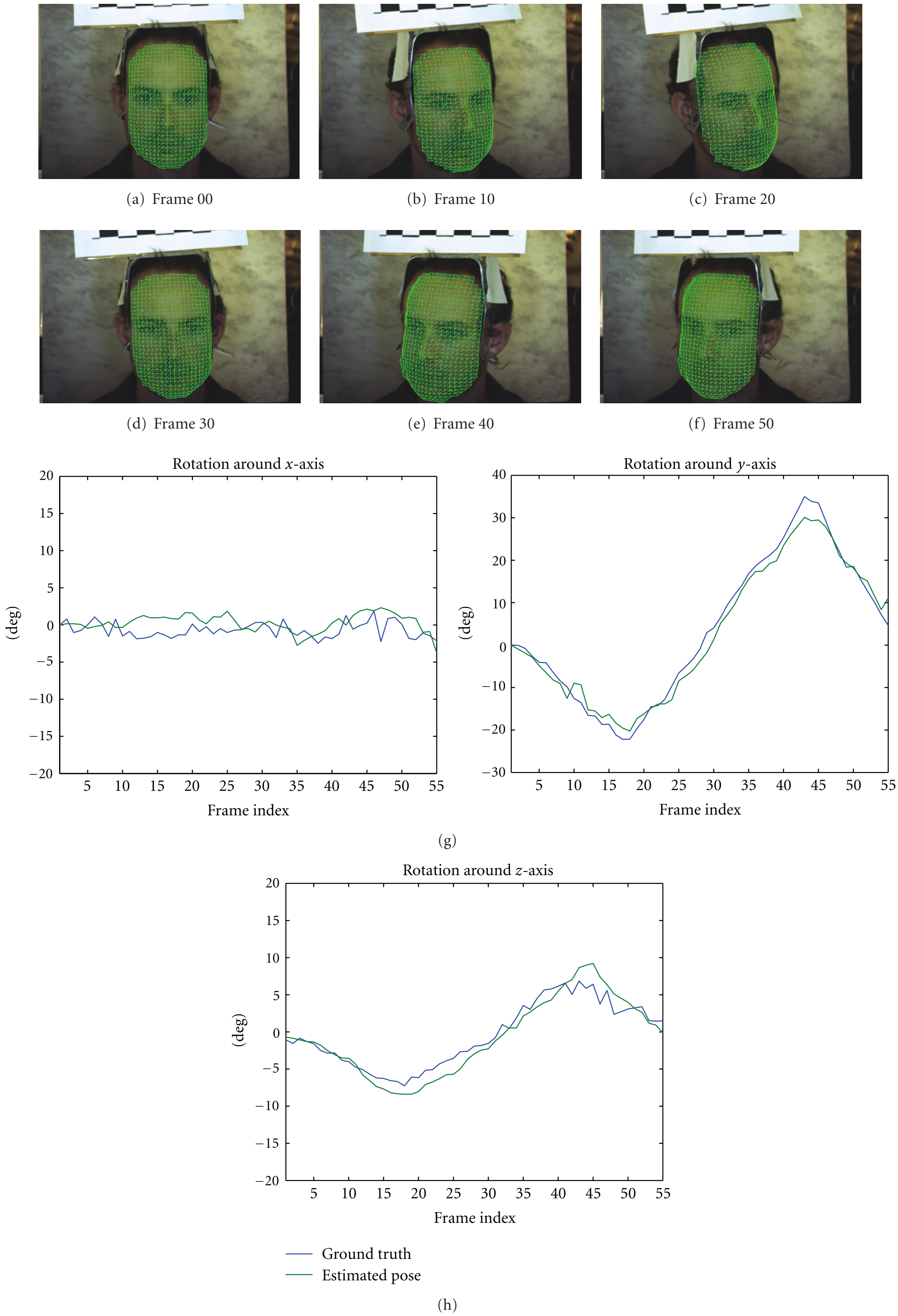


FIGURE 6: The estimated rotation around x-, y-, and z-axis of our sequences. The top rows show the recovered pose in some images. The bottom row is the estimated rotation.

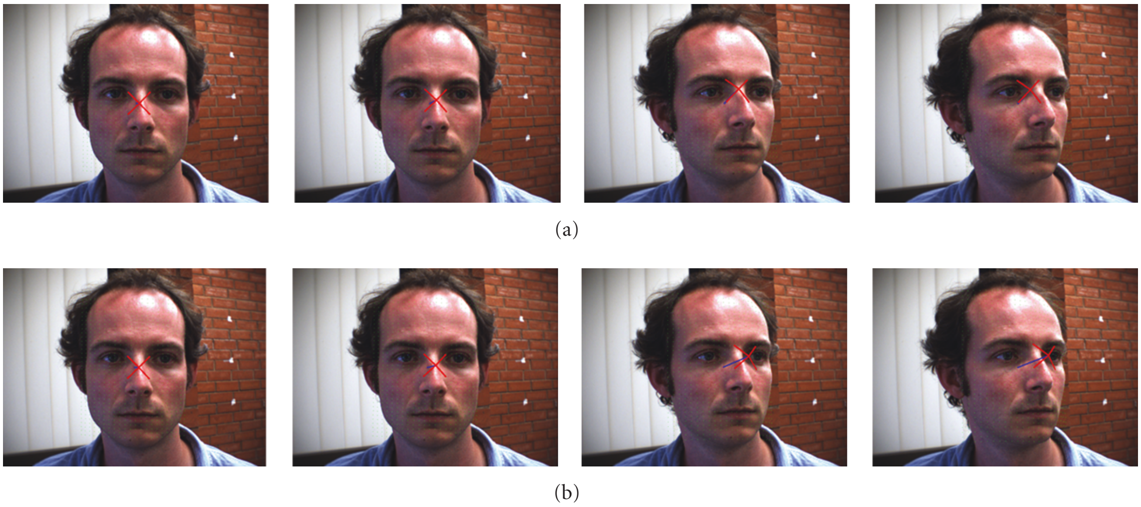


Figure 7: Comparison of intensity-based tracker, and hybrid tracker. (a) is the intensity tracker, and (b) is the hybrid tracker for the same sequence. The intensity-based tracker is more sensitive to the strong reflection.

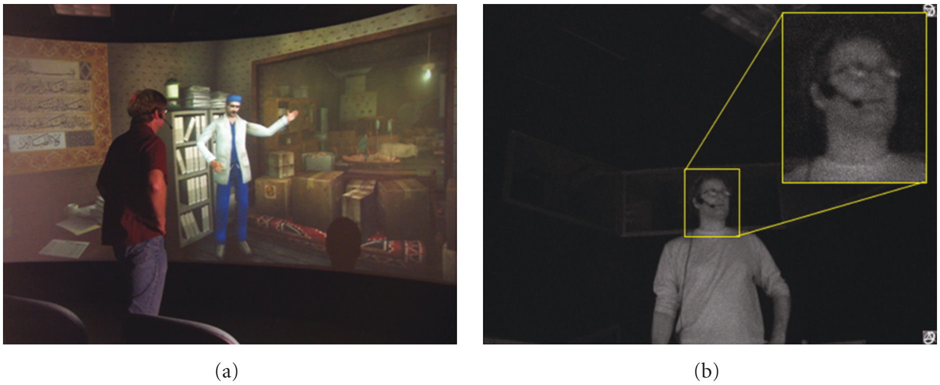


Figure 8: (a) Theater environment for head-tracking application. The subject is in nearly complete darkness except for the illumination from the screen. mage courtesy of USC's Institute for Creative Technologies. (b) Images from high-resolution IR camera placed below the screen.

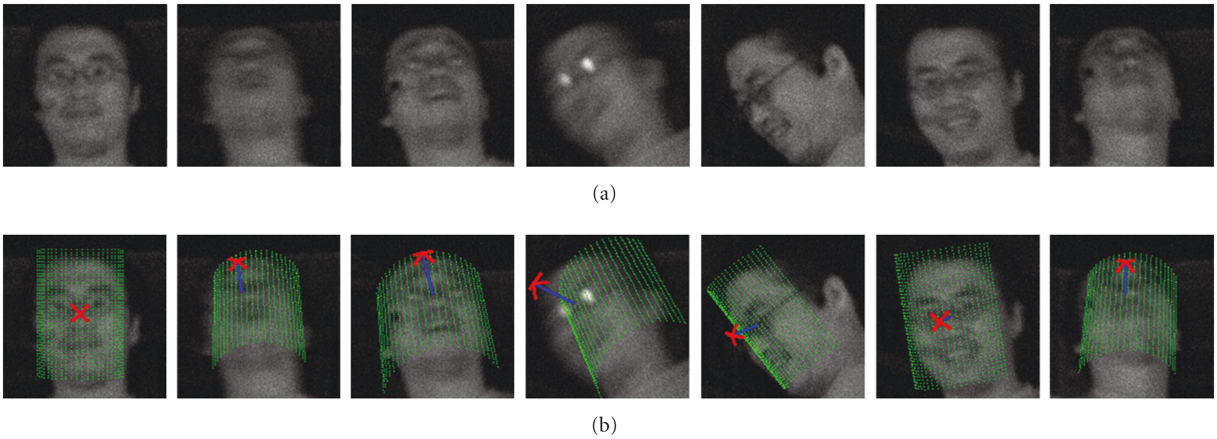


Figure 9: (a) shows some example frames, and the (b) row shows the estimation of the proposed tracker. The arrow indicates the direction that the user is facing. The feature-based tracker lose track completely in only 3 frames.

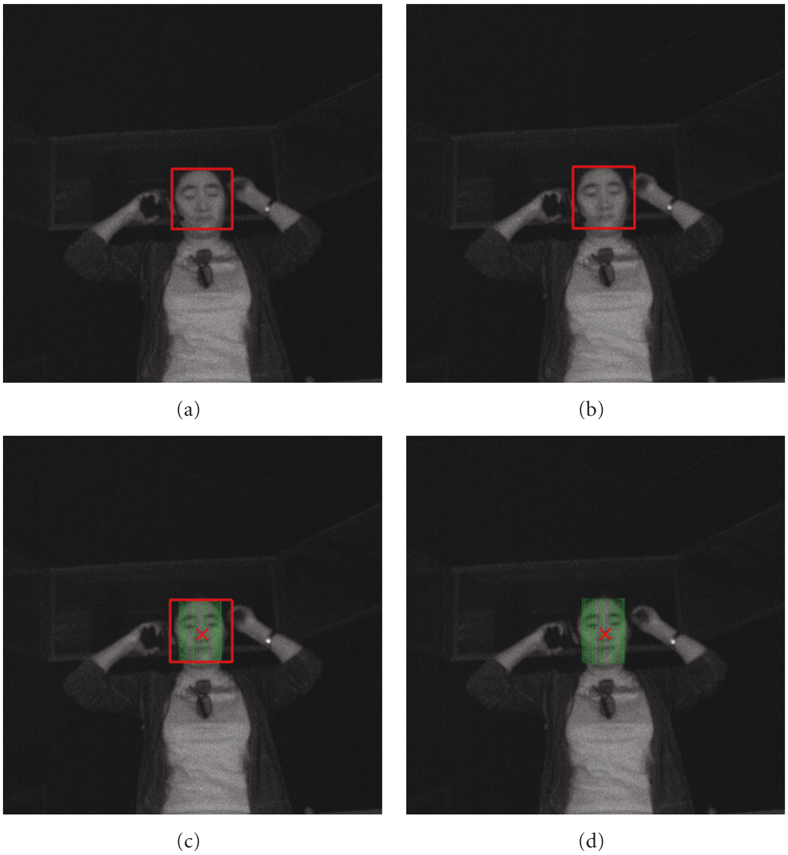


Figure 10: Automatic initialization of 3D head tracker. It shows 4 consecutive frames for automatic 3D head-tracker initialization. The red rectangle indicates the detected face region.

of initialization, we also use a 2D active shape model [19] to validate the existence of a frontal face, locate the semantic facial features, such as eyes and mouth corners, and align the 3D model to fit these features. Once we initialize the tracker, the proposed hybrid-tracking algorithm is used to estimate 3D head pose in subsequent frames.

The reliability of tracked pose is monitored by examining residual error of optimization and checking the possibility of estimated head pose. If the tracking is not reliable, the system switches to a reacquisition mode and turns on the face detector. Once a frontal face is located and tracker estimation is far away from the frontal pose, the tracker is reinitialized by using only the keyframe of the frontal pose, which is the first keyframe in our implementation, for optimization.

Figure 10 demonstrates the process of the automatic initialization. The detected face region is annotated by a red rectangle in Figure 10. The 3D head tracker is initialized only after the detector can continuously locate a face in a fixed position for several consecutive frames.

Figure 11 shows the reacquisition of 3D head tracker. As we showed in previous sections, current 3D head tracker is reliable near the frontal pose, but the accuracy decreases when the head approaches the extreme pose and is far away to the frontal view. In Figure 11, the accuracy of head tracker decreases from frame 710, as the subject approaches to a side view. The tracker is considered as "lost track" in frame 765 since the head pose is very different from the actual pose and the residual error becomes high. Thus, the tracker switches to a reacquisition model and searches for a frontal face. In frame 766, the reacquisition module detects a frontal face and uses it to reinitialize the 3D head tracker. After reacquisition, the tracker backs into the normal mode using hybrid-tracking algorithm.

6. Summary

We have presented a hybrid-tracking algorithm for robust real-time 3D face tracking. Built on a nonlinear optimization framework, the tracker integrates intensity information and feature correspondence for 3D tracking. Extensive empirical evaluation demonstrates that indeed the feature and intensity information is complementary and leveraging both achieves better accuracy than either alone. Several areas of investigation are opened up by this research. There are many options for the weighting scheme used during optimization. For example, the spatial distribution of features on the face will affect the overall model fit. If features are all clustered in one area, the resulting estimate will be less robust. This knowledge may be used to further adapt the optimizer to the available information.

In the future, we plan to use this tracker in several applications. One such application is for HCI, such as

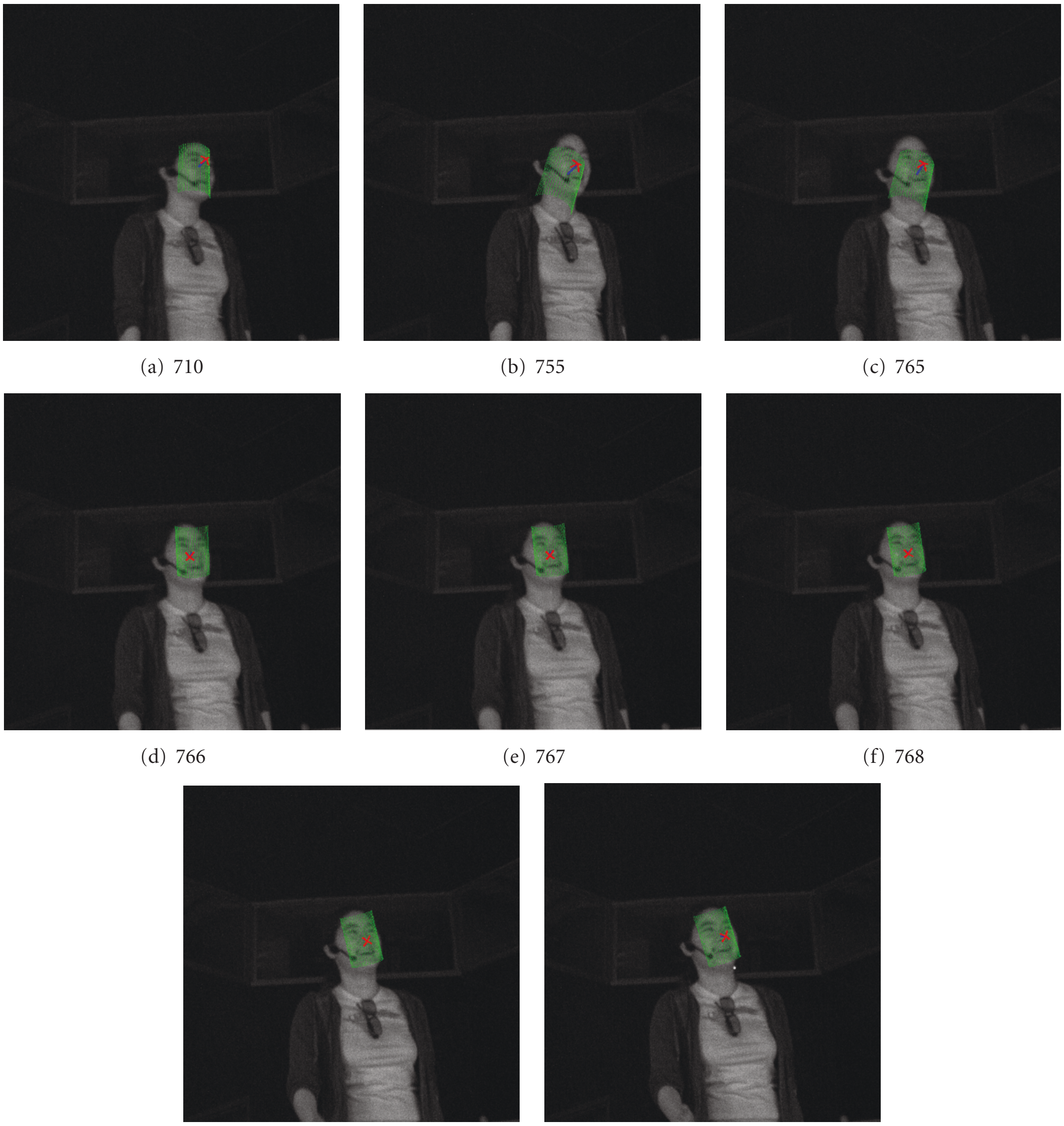

(a) 710 (b) 755 (c) 765 (d) 766 (e) 767 (f) 768 (g) 770 (h) 774

Figure 11: Automatic reacquisition of 3D head tracker. The number indicates the frame index.

in the theater environment presented in Section 5.3. The challenge here is stability on very long infrared sequences. We have applied the online keyframe generation technique to improve the stability, but the reliability of the generated keyframe remains an issue. The generated keyframe should be updated as the tracker gathers more information about the subject's face. Another problem is reinitialization. The current tracker has been shown to be robust under moderate facial deformation, thus has the potential for facial gesture analysis. Combining with a deformable model may improve the tracking accuracy and extend the ability to track nonrigid facial features.

References

[1] L. Vacchetti, V. Lepetit, and P. Fua, "Stable real-time 3D tracking using online and offline information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 10, pp. 1385–1391, 2004.

[2] J. Xiao, T. Moriyama, T. Kanade, and J. F. Cohn, "Robust full-motion recovery of head by dynamic templates and re-registration techniques," *International Journal of Imaging Systems and Technology*, vol. 13, no. 1, pp. 85–94, 2003.

[3] Y. Shan, Z. Liu, and Z. Zhang, "Model-based bundle adjustment with application to face modeling," in *Proceedings of the International Conference on Computer Vision (ICCV '01)*, vol. 2, pp. 644–651.

[4] D. Fidaleo, G. Medioni, P. Fua, and V. Lepeti, "An investigation of model bias in 3d face tracking," in *Proceedings of the IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, pp. 125–139, 2005.

[5] M. J. Black and Y. Yacoob, "Recognizing facial expressions in image sequences using local parameterized models of image motion," *International Journal of Computer Vision*, vol. 25, no. 1, pp. 23–48, 1997.

[6] S. Basu, I. Essa, and A. Pentland, "Motion regularization for model-based head tracking," in *Proceedings of the International Conference on Pattern Recognition (ICPR '96)*, vol. 3, pp. 611–616, 1996.

[7] L.-P. Morency, A. Rahimi, N. Checka, and T. Darrell, "Fast stereo-based head tracking for interactive environment," in *Proceedings of the Conference on Automatic Face and Gesture. Recognition (FGR '02)*, pp. 375–380, 2002.

[8] L.-P. Morency, A. Rahimi, and T. Darrell, "Adaptive view-based appearance models," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '03)*, vol. 1, pp. 803–810, June 2003.

[9] M. L. Cascia, S. Sclaroff, and V. Athitsos, "Fast, reliable head tracking under varying illumination: an approach based on registration of texture-mapped 3D models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 4, pp. 322–336, 2000.

[10] A. Schodl, A. Haro, and I. Essa, "Head tracking using a textured polygonal model," in *Proceedings of the Perceptual User Interfaces Workshop (held in Conjunction with ACM UIST 1998)*, 1998.

[11] D. DeCarlo and D. Metaxas, "Integration of optical flow and deformable models with applications to human face shape and motion estimation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '96)*, pp. 231–238, June 1996.

[12] L. Lu, X. Dai, and G. Hager, "Efficient particle filtering using RANSAC with application to 3D face tracking," *Image and Vision Computing*, vol. 24, no. 6, pp. 581–592, 2006.

[13] S. Baker and I. Matthews, "Lucas-Kanade 20 years on: a unifying framework," *International Journal of Computer Vision*, vol. 56, no. 3, pp. 221–255, 2004.

[14] S. Baker, R. Patil, K. Man Cheung, and I. Matthews, "Lucas-kanade 20 years on: part 5," Technical Report CMU-RI-TR-04-64, Robotics Institute, Carnegie Mellon University, Pittsburgh, Pa, USA, November 2004.

[15] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[16] FaceVision200. Geometrix, http://www.geometrix.co.uk.

[17] Photomodeler, http://www.photomodeler.com.

[18] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[19] L. Zhang, H. Ai, and S. Lao, "Robust face alignment based on hierar-chical classifier network.," in *Proceedings of the International Workshop on Human-Computer Interaction (HCI/ECCV '06)*, 2008.